

## IA54

### Final A2016

Tous documents autorisés,  
Tous dispositifs électroniques interdits  
Le barème est donné à titre indicatif (±1)

### 1 copie par exercice

#### Exercice 1 (7 points – N. Gaud) – MyPress : Application de revue presse personnalisée.

L'application MyPress est une application mobile accessible depuis tout terminal mobile et permettant :

- d'extraire/sélectionner des articles de presse en ligne depuis différentes sources de presse disponibles sur Internet (Reuters portal , The The New York Times, AFP, etc) et dans différents formats (news, articles, RSS, blogs, etc.). Les informations sont extraites, encodées dans un format standard facilitant la catégorisation du texte et son indexation, et sont stockées dans une base de données partagée pour tous les utilisateurs de l'application. Chaque utilisateur peut ajouter des sources d'information complémentaires s'il le désire.
- de classer automatiquement les informations disponibles en fonction des préférences de l'utilisateur considéré. Pour chaque utilisateur, une classification personnelle sera donc construite permettant de filtrer et hiérarchiser les informations disponibles dans la base précédente en fonction de ses préférences personnelles.
- de fournir des mécanismes de feedback permettant à l'utilisateur de paramétrer ses préférences en terme de contenu (liste de thèmes/sujets/catégories d'intérêts) et leurs évolutions dans le temps :
  - en choisissant a priori ses centres d'intérêts
  - mais également au fur et à mesure de l'utilisation de l'application en permettant notamment de taguer un article en précisant : *plus ou moins d'article de ce type*.Le système adaptera donc ses propositions d'articles en intégrant les retours de l'utilisateur.
- de fournir à l'utilisateur une liste de contenus personnalisés, d'articles de presse sous forme de listes classées par catégories. L'utilisateur pourra lire le contenu d'un article par simple clic sur l'article correspondant dans la liste (comme cela est fait par exemple dans les applications Google Actualités ou Google Play Kiosque).

**Question 1 (4 points) :** Décrire à l'aide du langage CRIO le modèle organisationnel de l'application MyPress en accord avec les éléments décrits ci-dessus : les différentes organisations, les rôles qui les composent et les capacités associées, les ressources externes (ex : BDD, GUI, etc.) si nécessaires ainsi que les liens potentiels entre ces organisations (Stereotyped UML Class diagram).

**Question 2 (3 points) :** Décrire un exemple d'instanciation de ce système présentant les différents agents logiciels nécessaires, les rôles qu'ils jouent au sein des différents groupes instanciant les organisations décrites à la question précédente (Cheeseboard diagram).

# IA54

## Final A2016

Tous documents autorisés,  
Tous dispositifs électroniques interdits  
Le barème est donné à titre indicatif (±1)

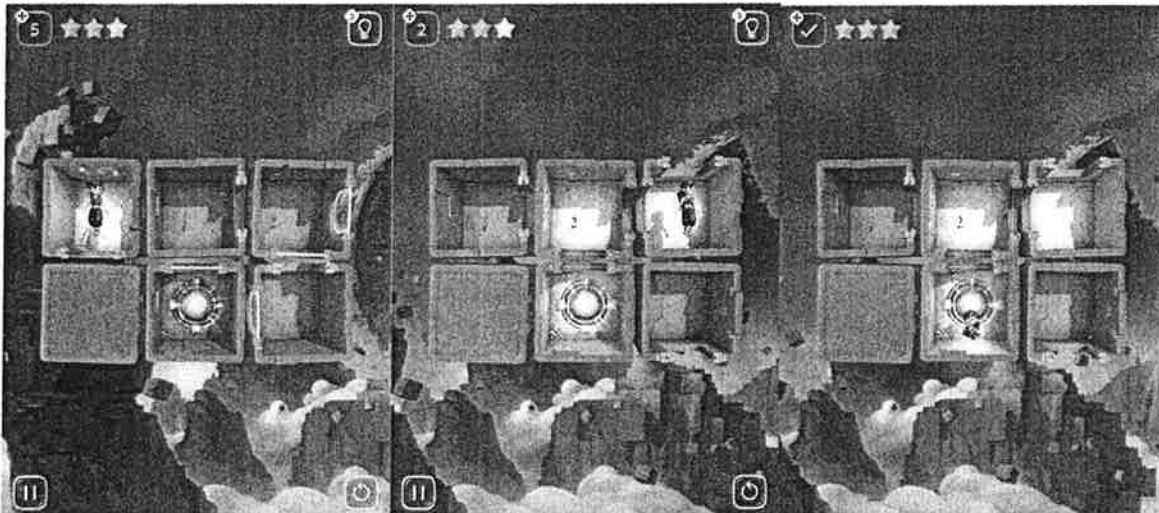
### 1 copie par exercice

#### Exercice 2 (5 points – F. Gechter) – Agents Réactifs.

**Question 1. question de cours :** Quelle sont les 2 grandes inspirations pour les modèles d'interaction des agents réactifs ? Citez deux exemples de chaque.

#### 2. Warp Shift

Warp Shift est un jeu sur plateforme mobile dont le but est de faire passer un personnage d'une position initiale à une porte menant vers le niveau suivant. Pour cela, on peut faire passer le personnage de case en case si celles-ci sont reliées par une porte de la même couleur. Il est également possible de faire bouger les cases verticalement ou horizontalement. Lors d'un déplacement celles-ci vont faire une permutation circulaire. Chaque action du joueur (déplacement du personnage d'une case ou déplacement, vertical ou horizontal, des cases) compte pour une action. La résolution du problème doit se faire en un nombre limité d'action, ce nombre étant déterminé en fonction du niveau à résoudre. Les images suivantes donnent un aperçu du jeu. L'image de gauche donne l'état initial. Celle du milieu donne un état intermédiaire obtenu en faisant une permutation circulaire vers la gauche de la première ligne, un déplacement du personnage vers le bas, et une permutation circulaire de la dernière colonne. L'image de droite correspond à la configuration finale désirée.



**Question 2.1 :** Parmi les méthodes de résolution de problème à base d'agents réactifs que nous avons vu en cours, laquelle semble la plus appropriée pour résoudre ce problème ?

**Question 2.2 :** En utilisant cette méthode définissez les différents éléments nécessaires pour l'appliquer à ce jeu. Quels sont les agents ?, quels sont leurs comportements ?, leurs interactions ?,...

## IA54

### Final A2016

Tous documents autorisés,  
Tous dispositifs électroniques interdits  
Le barème est donné à titre indicatif ( $\pm 1$ )

#### 1 copie par exercice

#### Exercice 3 (4 points – F. Lauri) – Apprentissage par renforcement.

Soit le Processus Décisionnel de Markov (PDM)  $M=(S,A,T,R,\lambda)$  défini par :

- .  $S = \{ 1, 2, 3 \}$  l'ensemble des états ;
- .  $A = \{ a_1, a_2, a_3 \}$  l'ensemble des actions ;
- .  $T = \{ (1,a_1,2), (1,a_2,2), (1,a_3,1), (2,a_1,3), (2,a_2,1), (2,a_3,3), (3,a_1,1), (3,a_2,2), (3,a_3,3) \}$  l'ensemble des transitions entre états ;
- . R la fonction de récompense telle que :
  - .  $R(s,a) = 3$  si  $(s,a,3) \in T$  (autrement dit si l'application de l'action a dans l'état s mène à l'état 3).
  - .  $R(s,a) = 1$  si  $(s,a,2) \in T$
  - .  $R(s,a) = 0$  sinon ;
- . et  $\lambda = 0,9$ .

**Question 1 (1 point) :** Pourquoi l'algorithme Q-Iteration est particulièrement adapté pour résoudre la tâche représentée par le PDM M ?

**Question 2 (2 point) :** Appliquez 3 itérations de Q-Iteration en indiquant à chaque itération la matrice des Q-valeurs mise à jour ainsi que les choix éventuels réalisés par l'algorithme.

**Question 3 (0,5 point) :** Donnez la politique  $\pi$  obtenue à partir de la matrice finale des Q-valeurs.

**Question 4 (0,5 point) :** Cette politique est-elle optimale ? Justifiez votre réponse.

## IA54

### Final A2016

Tous documents autorisés,  
Tous dispositifs électroniques interdits  
Le barème est donné à titre indicatif (±1)

#### 1 copie par exercice

#### **Exercice 4 (4 points – S. Galland) – Plateformes Multi-Agents.**

Un agent doit pouvoir migrer d'un micro-noyau vers un autre micro-noyau. La migration d'un agent consiste à copier l'état de l'agent (attributs) sur un autre micro-noyau, puis de supprimer l'original. Cette migration est transparente et non invasive pour l'agent. Pour cela, l'agent doit disposer de suffisamment de ressources sur le noyau distant avant de pouvoir migrer. Il est par conséquent nécessaire que le micro-noyau de départ garantisse la disponibilité des ressources à l'agent. Pour cela, il doit communiquer avec l'autre micro-noyau pour obtenir la quantité de ressources disponibles. Nous considérerons dans cet exercice que la mémoire est toujours disponible. En d'autres termes, il y a toujours assez de mémoire.

La fonction demandant la migration d'un agent à un micro-noyau est (syntaxe SARL) :

```
def migrateTo(remoteKernel : Address, agent : Agent) : boolean
```

Le paramètre remoteKernel est l'adresse du noyau destination. Vous supposerez que l'agent connaît cette adresse. Le paramètre agent est la référence sur l'agent voulant migrer. La fonction migrateTo retourne une valeur booléenne indiquant si la migration est autorisée par le micro-noyau.

La fonction suivante permet à un micro-noyau d'obtenir la taille de mémoire nécessaire pour un agent (syntaxe SARL) :

```
def computeAgentSize(agent : Agent) : long
```

La fonction suivante permet à un micro-noyau d'obtenir la taille de la mémoire libre sur son calculateur (syntaxe SARL) :

```
def getSystemFreeMemory : long
```

**Question 1 :** Écrivez un diagramme de classe décrivant les éléments constituant la plate-forme et uniquement nécessaires au processus de migration. Vous ignorez les éléments de la plate-forme n'intervenant pas dans ce processus.

**Question 2 :** Écrivez un diagramme de séquence décrivant le processus de migration d'un agent. Les entêtes des lignes de vie seront les éléments de votre diagramme de classe. Les messages décrits entre les lignes de vie seront les messages échangés ou les appels de fonctions existants entre les micro-noyaux et/ou l'agent voulant migrer.